Particle Swarm Optimization Applied to Intelligent Vehicles Squad Coordination

Gustavo Pessin * **Fernando S. Osório** * **Denis F. Wolf** *

* *Institute of Mathematics and Computer Science (ICMC)*
University of São Paulo (USP)
Av. Trabalhador São-Carlense, 400 – São Carlos, SP, Brazil
(e-mail: {pessin, fosorio, denis}@icmc.usp.br)

Abstract: This paper presents the modeling, implementation and evaluation of the Particle Swarm Optimization (PSO) applied to intelligent vehicles group formation and coordination. The robotic task discussed in this paper is performed over a natural disaster scenario, simulated as a forest fire. The intelligent vehicles squad mission should surround the fire and avoid fire's propagation. Experiments have been carried out with several PSO parameter's variation (e.g. inertia, confidence, social models, swarm size) seeking to get the more efficient optimization for the formation of the group. This paper describes all performed experiments detailing all sets of parameters, including positive and negative results. The simulation's results showed that with an adequate set of parameters is possible to get satisfactory strategic positions for a multirobotic system's operation using PSO.

Keywords: Particle swarm optimization, multirobotic system, coordination.

1. INTRODUCTION

There are many fields where a single agent is not sufficient or enough to fulfill a task. Tasks like cleaning nuclear residuals, cleaning chemical accidents, forest fire combat or even on constructions, agriculture, hostile environment exploration, security and critical missions may be better accomplished when using a group of agents. Using robotic agents (as intelligent vehicles) instead of human beings may add security, reliability and efficiency in these tasks. Multirobotic systems are extremely dependent on control techniques; they can add mobility, flexibility and robustness to a wide range of new applications (Mondada et al., 2005), but they also bring a series of new questions to be solved in collaboration and cooperation. Specialized algorithms, composed by rules and automats have been developed seeking to coordinate these physical sets in dynamic environments, showing to be an extremely complex challenge (Go et al., 2004). Due to it, a large number of researchers are migrating their efforts to several different approaches (e.g. application of classical intelligent artificial techniques, social models, market-based models, swarm-based models).

In the firefighting mission, one of the most important questions is related to the robot position setting. According to the actuation capability of each robot, weather condition (wind, rain), topography, and vegetation, several arrangements can be proposed. These arrangements, when suggested by a specialist, may not take in account a large number of variables, which makes it a hard task. In these cases, machine learning techniques may be succesfully used. One of the machine learning technique that has been showing satisfactory results in solving optimization problems is Particle Swarm Optimization, which is a stochastic technique inspired by social behaviors (Kennedy and Eberhart, 1995). Monitoring and combating of forest fire is an example of multirobotic system that could considerably reduces human, material and environmental losses.

In (Pessin et al., 2007) we proposed a simulation environment for recognition and combat of forest fire with rule-based agents. In (Pessin et al., 2010) we presented an evolution of the simulation environment, using the physical simulation library Open Dynamics Engine (ODE) (Smith, 2009), where new artificial intelligence techniques were applied on the agents (Artificial Neural Networks and Genetic Algorithms). The present work's goal is to describe experiences with Particle Swarm Optimization performing a search for optimal acting positions in a multirobotic system [1]. We evaluate parameters like inertia, confidence, types of social models and swarm size, considering a total of 54 different sets of parameters. The PSO evaluation is done considering a robotic task performed over a natural disaster, simulated as a forest fire propagation. Experiences with two and four firefighter robots have been done.

This paper has the following structure: Section 2 introduces short theoretical description of robot's [2] applications. Section 3 presents concepts and applications of Particle Swarm Optimization. In section 4 we explain the developed environment, the proposed fitness and the particle's structure. Section 5 describes the evaluation of all performed experiments. We finalize presenting the conclusion of the presented work and the future perspectives.

2. MOBILE ROBOTICS

Several current works demonstrate mobile robotic usage (individual system) on hostile operations as the rescue

[1] Source-code available at http://sites.google.com/site/pessin
[2] We use the term robot as a synonym for intelligent vehicles

auxiliary robot Raposa (IdMind, 2009) and SACI robot (Macedo et al., 2007) developed for acting on fire combat. Moreover, there are robots to perform tasks on aquatic environments, space, caves and volcanoes exploration, and even to household use. Multirobotic systems must be formed by robots that are able to effective act on tasks, so knowledge about robotic control is a very important field. Works describing intelligent robot navigation can be seen in (Zhao and Collins, 2005; Heinen et al., 2006). In 2004 and 2005, DARPA Grand Challenge (Darpa, 2007), financed by the Defense Advanced Research Projects Agency organized a competition where the goal was building a completely autonomous vehicle that could complete a long way on dirt road on limited time. In 2007 the focus of the competition has changed. Renamed to DARPA Urban Challenge, had a new goal to build a vehicle that could be autonomous on urban traffic, and realize tasks like parking, overtaking and intersection negotiations. These examples show trends in cooperation and multiple interactions.

The work with groups adds a great number of possibilities on tasking-solving but bring a series of new questions to be solved in collaboration and cooperation. Works using multirobotic systems like (Yamaguchi, 1997; Balch and Arkin, 1998) uses pre-programmed rules on agents to perform formation. In (Mondada et al., 2005; Dorigo et al., 2004) are explored techniques to perform works with collectives robotics, used mainly for the purpose of applying the concept of self-organization and collective optimization, but task division is not directed explored. The works described in this section demonstrate that the application of mobile robotics in control of incidents is an important and active topic of research and development. These several competitions also demonstrate that there is still not a definitive or more adequate solution to the problem, and it is an open research field. In all consulted documents there is no consensual form to multirobotic system's conformation and actuation. Unpredicted situations with large degree of autonomy and robustness are still difficult to handle.

3. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) is a stochastic optimization technique, inspired by social behavior of bird flocking and fish schooling (Eberhart et al., 2001; Engelbrecht, 2005). The optimization process occurs in two simultaneous ways: through cooperation (group learning) and competition (individual learning) among particles (individuals) from a swarm (population).

PSO shares many concepts with evolutionary computation techniques such as Genetic Algorithms (GA), where there is an initial population (where each individual represents a possible solution) and a fitness function (whose value represents how far an individual is to an expected problem's solution). However, unlike GA, PSO has no explicit concepts of evolution operators such crossover or mutation. In the PSO, there is a swarm of randomly created particles. On each algorithm iteration, each particle is updated following: (i) best population fitness (ii) best fitness found by the particle (considering past generations of the particle). Each particle has a position x (or a position vector) and a velocity v (or velocity vector). The position represents a solution for the problem and the velocity defines the particles displacement direction weight.

New particle's position is given by Eq. 1. Where x_k^i is the position of particle i at instant k and v_k^i is particle's i velocity at k moment. Particle's velocity is updated in accord to Eq. 2.

$$x_{k+1}^i = x_k^i + v_{k+1}^i \tag{1}$$

$$v_{k+1}^i = w \cdot v_k^i + c_1 \cdot r_1(pbest - x_k^i) + c_2 \cdot r_2(gbest - x_k^i) \tag{2}$$

On Eq. 2, v_k^i is particle's actual velocity, w represents a particle inertia parameter, *pbest* is the best position among all positions found by the individual (particle best), *gbest* is the best position among all positions found by the group (group best), c_1 and c_2 are trust parameters, r_1 and r_2 are random numbers between 0 and 1. Parameters (w, c_1, c_2, r_1 e r_2) are detailed in the sequence.

The velocity is the optimization's process guide parameter (Engelbrecht, 2005) and reflects both particle's individual knowledge and group knowledge. Individual knowledge is known as *Cognitive Component* and group knowledge is known as *Social Component*. Velocity consists of a three-term sum: (i) Previous speed: used like a displacement direction memory and can be seen as a parameter that avoids drastic direction changes; (ii) Cognitive Component: directs individual to the best particle's found position so far, resembles to best position individual memory of the particle; (iii) Social Component: directs individual to the best group's position found.

Parameters c_1 and c_2, also called trust, are used to define individual or social tendency importance. Standard PSO works with static and equal trust values (c_1=c_2), that means that the group experience and the individual experience are equally important (called *Full Model*). When c_1 parameter is zero and parameter c_2 is higher than zero, PSO uses only group information (called *Social Model*). When parameter c_2 is zero and parameter c_1 is higher than zero, PSO uses only particle's information, disregarding group experience (called *Cognitive Model*). Random value introduction (r_1 and r_2) on velocity adjust allows PSO to explore on a better way several search space points (Engelbrecht, 2005). Inertia parameter aims to balance local search or global search. As the value approximate to 1.0, search gets close to global search, while lower values allow local search. Usually this value is between 0.4 and 0.9. Some authors (Eberhart and Shi, 2001; Kennedy and Eberhart, 1995) suggest value's linear decay, but warn that linear decay use is not always the best solution. Most parameters are problem-dependent (Engelbrecht, 2005).

PSO is used in general where Evolutionary Computing can also be used. Examples of works that describe comparison between the two techniques are (Eberhart and Shi, 1998; Eberhart et al., 2001; Engelbrecht, 2005). The work (Pugh and Martinoli, 2006) describes a PSO development for robot navigation control where distance sensor values are used as input information. The work (Rong et al., 2008) describes the development and evaluation of a PSO for four legs robot walking; the PSO optimize time and intensity of force application on several joints. Similar approaches can be found in (Niehaus et al., 2007), but in this case bipedal robots were the case of study. These works achieved satisfactory results for static environment. One work that explores dynamic operations can be seen on (Burchardt and Salomon, 2006), whose uses a fault monitor. The system has two phases: planning and action, when the system is on action mode and identifies a possible failure, it automatically stops and reactivates the planning mode. This allow the application of the system in dynamic environments.

4. GROUP FORMATION

In order to build a real physical implementation of robotic system, it is highly recommended to test the algorithms on virtual realistic simulation environments. Robotic system's simulation is specially necessary in case of big, expensive or fragile robots because it is an powerful tool to avoid wasting resources (Go et al., 2004). In our case, the proposed simulator should be able to reproduce an environmental disaster for a multirobotic system actuation. We propose the situation of a forest fire, so, in this case a intelligent vehicle squad (as road grader) has the purpose of combat the forest fire acting by creating firebreaks around the fire. The developed 3D simulation environment uses the OSG library (OSG, 2009) which is responsible by graphic output, the Demeter library (Demeter, 2009) that is responsible by irregular terrain generation and ODE library (Smith, 2009) which is responsible by physic realism, both in the robotic morphology as in the collision involving the objects presents in the environment (e.g. robots, trees, terrain inclination). Using ODE library allows the physically simulated robots to comply with gravity, inertia and friction. Also, a 2D simulation environment was build to allow faster simulations (ignoring physical restrictions on robot navigation). This 2D prototype is implemented with SDL (SDL, 2009), as shows Fig. 3. Both the prototypes have the same fire propagation comportment and use C/C++ as programming language.

The control variables of vegetation simulation and fire propagation are updated by a hidden matrix under the terrain. This matrix has the type of present vegetation for each terrain region; consequently, associating this information with the wind orientation and intensity we can build the fire propagation simulation. Regarding the wind, both its intensity and its orientation can be generated randomly or configured with parameters defined by the user. The fire remain in an area related directly to the present vegetation type and behaves in basis of terrain type values, terrain slope, wind orientation and intensity. In this way the fire spreading simulation try to model the fire propagation as realistic as possible. To implement the fire spreading, we obtained from (Koproski, 2005) real velocity measurements. The detailed characteristics about fire spreading modeled to this work, as well as the forest fuel models and the real operation techniques are compiled into (Pessin, 2008). The simulated terrain is based on topographical maps and on forest fuel maps models that can also be seen in (Pessin, 2008).

The 3D simulation tool uses a realistic autonomous vehicle model, defining a 4 wheel vehicle (Fig. 4(d)) with steering and acceleration/break control (physically simulated: vehicle kinematics and dynamics controlled by ODE tool).

Table 1. Particle structure (group of four vehicles - angle and radius related to the fire's starting point).

Sub-particle	Function	Min. value	Max. value
0	Initial angle of vehicle 0	0.0º	360.0º
1	Final angle of vehicle 0; initial of vehicle 1	0.0º	360.0º
2	Final angle of vehicle 1; initial of vehicle 2	0.0º	360.0º
3	Final angle of vehicle 2; initial of vehicle 3	0.0º	360.0º
4	Final angle of vehicle 3	0.0º	360.0º
5	Initial radius of vehicle 0	10.0m	100.0m
6	Final radius of vehicle 0; initial of vehicle 1	10.0m	100.0m
7	Final radius of vehicle 1; initial of vehicle 2	10.0m	100.0m
8	Final radius of vehicle 2; initial of vehicle 3	10.0m	100.0m
9	Final radius of vehicle 3	10.0m	100.0m

The vehicle was configured with laser simulated sensor used in order to detect and avoid obstacles. The vehicle is controlled by an Artificial Neural Network that reads the sensors (laser), estimaded position/orientation (GPS, compass) and generates the commands to the actuators (steering/acceleration). More details about vehicle development can be seen in (Pessin et al., 2010).

The PSO algorithm optimizes fire combat position for each intelligent vehicle on group, specifically: (i) Initial combat position for each member of the group (beginning point of firebreak creation); (ii) Final combat position for each member of the group (final point of firebreak creation). These positions are send by command messages used to activate the robots. To perform the simulations is necessary: (i) Knowing the available number of robots; (ii) Knowing robot's operation speed; (iii) Knowing robot's initial position; (iv) Having the ability to simulate fire propagation. To simulate fire propagation is necessary: (i) Getting initial fire position; (ii) Getting wind direction; (iii) Getting a simplified copy of the map (land and vegetation). This set of proposed information can be obtained by sensors. The proposed particle's structure has information about the entire group of robots. Therefore, we need 10 values to a group of four vehicles. These values are stored in subparticles (position vector, such as genes in a GA). In the proposed structure of the particle, the final position of a robot is the starting position of the next, as shown on Tab. 1.

The coordinates of operation are calculated applying Eq. 3 and 4 to the best particle. Where (x_d, y_d) is the robot's destination position, (x_a, y_a) is the starting position of the fire, r_i is the radius (subparticle 5 to 9) and a_i is the angle (subparticle 0 to 4). The radius and the angle are specifics to each operation of each robot (initial and final coordinate of firebreaks creation).

$$x_d = x_a + r_i \cdot \cos(a_i) \tag{3}$$

$$y_d = y_a + r_i \cdot \sin(a_i) \tag{4}$$

In the implementation of the algorithm, we used a concept similar to the alleles on GAs, in order to reduce the search space. Thus, the radius can be between 10.0 and 100.0 units and the angle can be between 0.0º and 360.0º. The values stored in the particle are floating-point numbers. Also, we used star-type social structure; where all particles have connections to each other, in practical terms, that means there is a unique *gbest* for all particles. Star-type model has faster convergence compared with other structures (Engelbrecht, 2005; Kennedy, 1999). More details about parameter's variation are described in Section 5.

Table 2. Set of PSO initial evaluations.

Set	ConfP (c_1)	ConfG (c_2)	Inertia (w)	Particles
A	0.0	2.0	0.4	80
B	0.0	2.0	0.8	80
C	0.0	2.0	1.2	80
D	0.0	2.0	0.4	160
E	0.0	2.0	0.8	160
F	0.0	2.0	1.2	160
G	2.0	0.0	0.4	80
H	2.0	0.0	0.8	80
I	2.0	0.0	1.2	80
J	2.0	0.0	0.4	160
K	2.0	0.0	0.8	160
L	2.0	0.0	1.2	160
M	2.0	2.0	0.4	80
N	2.0	2.0	0.8	80
O	2.0	2.0	1.2	80
P	2.0	2.0	0.4	160
Q	2.0	2.0	0.8	160
R	2.0	2.0	1.2	160

The fitness function guides PSO optimization. The proposed fitness is related with saved vegetation area and combat units usage rate; therefore, the fitness accumulates: (i) Total burned area: trying to minimize burned area, (ii) Firebreak total area: trying to minimize robot's work area, avoiding to create firebreak on non-risk areas, (iii) Trying to minimize the difference among general average of useful firebreaks in relation to each individual useful firebreak, equalizing worked areas. The PSO tries to minimize the fitness function value, this at means less burned vegetation, less created firebreaks, and less difference between the size of firebreaks of each robot.

5. EXPERIMENTS AND RESULTS

Considering that convergence speed is one of the most important aspects for the proposed system, evaluations have been done to verify and find the best parameter set for the proposed PSO. Tab. 2 presents the initial parameter variation list. We fixed the climatic characteristics of the fire simulation and the initials robot positions in order to make the evaluations. Six simulations were executed for each parameter set, which implies on a total of 108 simulations. We made visual observations on 20% of the simulation results to verify the ideal fitness for fire combat.

Sets {A..F} have only confidence on group (social model, c_1=0.0 and c_2=2.0). Sets {G..L} have only confidence on particle (cognitive model, c_1=2.0 and c_2=0.0). Sets {M..R} have confidence on group and on particle (full model, c_1=2.0 and c_2=2.0). The Fig. 1 presents some simulation's visual results. Fig. 1(a) and 1(b) presents simulations sequences with a fitness of 3800 units; the fire is stopped but the firebreak is poorly optimized. Fig. 1(c) presents a fitness of 3480 units and Fig. 1(d) presents a fitness of 2643 units. The last two figures present visual-efficient operations but in mathematical terms the resulting particle analysis shows that on Fig. 1(c) the standard deviation over actuation areas average is of 6.05 degrees and 2.37 radius units and Fig. 1(d) standard deviation over actuation areas average of 3.38 degrees and 0.23 radius units. Both teams extinguished the fire but Fig. 1(c) shows more equalized work areas among robots. Thus, from visual observations, fitness under 3500 units were defined as ideal for this experiment.

From initial evaluations (Tab. 2) only {B,E} (social model with w=0.8) and {M,P} (full model with w=0.4) had at least 50% of results with fitness under 3500 units. No configuration had more than 70% of solutions with fitness under 3500 units. Thus a new evaluation set (Tab. 3 and 4) was performed considering the four types of variations that had the best results, with an increase on number of generations (from 500 to 800), particle amount (20, 50, 100 and 200) and inertia (+/- 0.1). The simulations using the cognitive model results significantly poorly values in relation to social models and full model. So, no evaluation with the cognitive model was made in the second round of evaluations.

Tab. 3 and 4 also presents 10 simulations results for each parameter set. The tables show that the only configuration

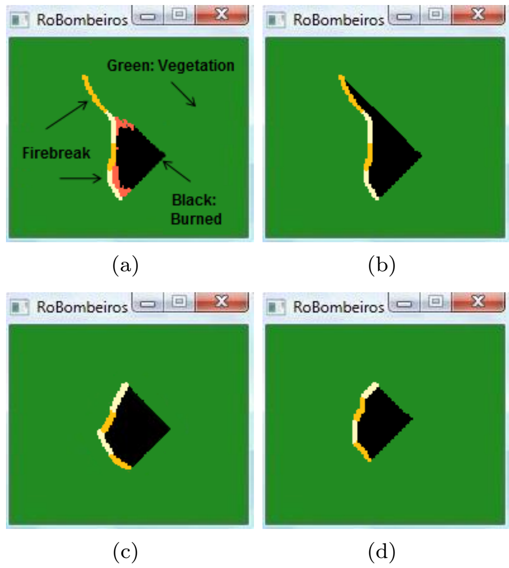


Fig. 1. (a) and (b) Sequences of a simulation which result in a fitness equal to 3800 units; the fire is stopped but the firebreak is poorly optimized. (c) Simulation which result in a fitness equal to 3480 units; (d) Simulation which result in a fitness equal to 2643 units.

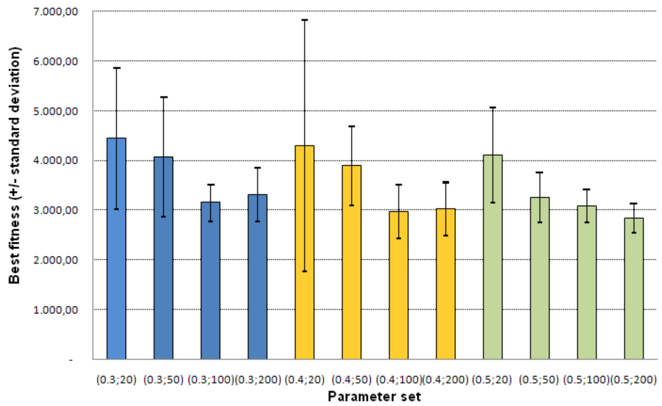
Table 3. New set of evaluations (social model).

Set	Swarm size	Inertia	Results with fitness $<$ 3500
S_0	20	0.7	0%
S_1	20	0.8	40%
S_2	20	0.9	0%
S_3	50	0.7	40%
S_4	50	0.8	70%
S_5	50	0.9	0%
S_6	100	0.7	60%
S_7	100	0.8	70%
S_8	100	0.9	0%
S_9	200	0.7	80%
S_{10}	200	0.8	80%
S_{11}	200	0.9	0%

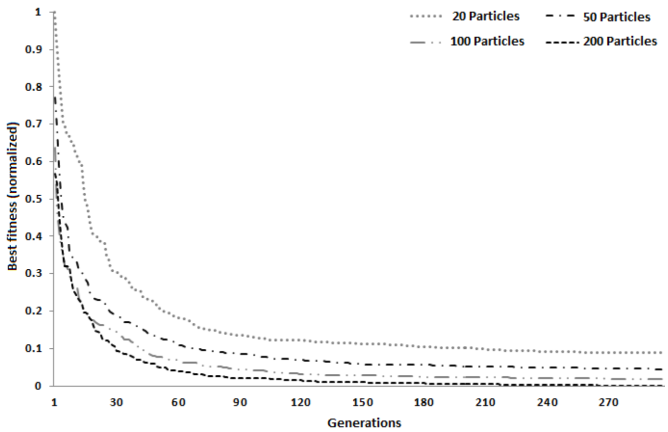
Table 4. New set of evaluations (full model).

Set	Swarm size	Inertia	Results with fitness $<$ 3500
F_0	20	0.3	40%
F_1	20	0.4	40%
F_2	20	0.5	40%
F_3	50	0.3	40%
F_4	50	0.4	30%
F_5	50	0.5	50%
F_6	100	0.3	80%
F_7	100	0.4	80%
F_8	100	0.5	80%
F_9	200	0.3	70%
F_{10}	200	0.4	80%
F_{11}	200	0.5	100%

that had 100% of the fitness results under 3500 units was the set F_{11}. The parameter set used on F_{11} is c_1=2.0, c_2=2.0 (full model), w=0.5, 800 generations and 200 particles. We can see on tables that the experiences with 20 and 50 particles presents weak results, in comparison with the experiences with 100 and 200 particles. This happens in full model as in social model. Also, we can see on social model (Tab. 3) that the experiments that use inertia=0.9 show unsatisfactory results, regardless of the amount of particles present in the system. Other information that can be seen from tables is that the experiments with the full model (Tab. 4) showed better results compared to the experiments on the social model (Tab. 3). The Fig. 2(a) presents best fitness average graphic from Tab. 4 (average of 10 simulations +/- standard deviation). In accord with presented on Tab. 4 set F_{11} had the lower average and lower standard deviation.



(a)



(b)

Fig. 2. (a) Results of the evaluations described in Tab. 4. The x-axis shows the description of the experiment as (inertia; swarm size). (b) Evolution of fitness according to number of generations and different swarm size (w=0.5).

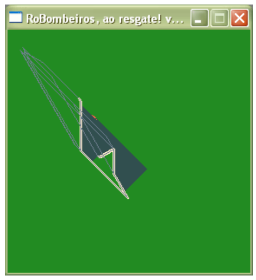
Fig. 2(b) presents fitness evolution graphic for four combat robots (10 simulations average - full model and inertia of 0.5). Fire spread simulation considered East-West wind direction and relative wind speed at 7km/h; robot navigation speed of 35km/h; robots positioned on 2km far from fire threshold base. Fig. 2(b) shows that the best fitness obtained is with a particle amount of 200. From 150 generations fitness optimization is almost stabilized.

Table 5. Best particles (resultant of three simulations).

	Simulation		
Subparticle	A	B	C
0	226.60	225.71	227.21
1	202.46	204.55	202.91
2	176.87	178.78	175.79
3	161.13	163.21	161.12
4	138.14	137.58	138.17
5	26.51	26.84	29.83
6	29.20	28.79	30.34
7	30.03	29.88	28.98
8	30.12	26.27	28.21
9	32.33	28.69	32.11

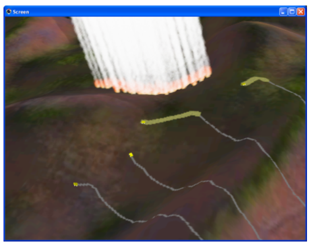
Best particles resulting from three simulations using described parameters can be seen on Tab. 5. Fig. 3 and 4 presents some simulations scenes results with satisfactory and unsatisfactory results.

(a)

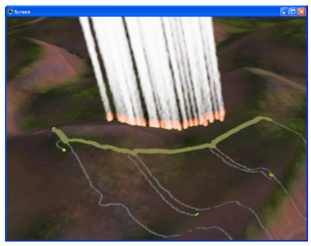


(b)

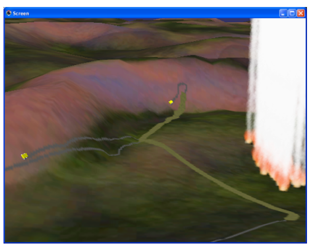
Fig. 3. Unsatisfactory results. (a) 20 particles, w=0.7 and social model: the fire is not contained by the firebreak. (b) 50 particles, w=0.3 and full model: the firebreak are too large relative to what would be necessary and are not well distributed among the robots.



(a)



(b)



(c)

(d)

Fig. 4. Satisfactory results of the PSO (set F_{11}). (a) and (b) Four vehicles creating a firebreak. (c) Two vehicles creating a firebreak. (d) Detailed view of navigation with obstacles avoidance.

Fig. 4 presents satisfactory evolution result applied on 3D virtual simulation environment. The 3D prototype showed that robots completely surround the fire and create the firebreaks on a satisfactory way. It's important to mention that some simulations using different navigation speed and fire propagation were performed but on a small number of rounds. So, they are not detailed on this text but also presented satisfactory results. For navigation in the irregular terrain (including obstacles avoidance using laser sensors) the robots use Artificial Neural Networks detailed in (Pessin et al., 2010).

6. CONCLUSIONS AND FUTURE WORK

This paper presents the modeling, implementation and evaluation of the Particle Swarm Optimization (PSO) applied to intelligent vehicles group formation and coordination. The robotic task discussed in this paper is performed over a simulated forest fire. Simulations have been carried out with several PSO parameter's variation (e.g. inertia, confidence, social models, swarm size) seeking to get the more efficient optimization for the formation of the group. The simulation's results showed that with an adequate set of parameters it is possible to get satisfactory strategic positions for a multirobotic system's operation using PSO.

Some approaches are planned as future work: (i) a detailed study on others methods for robot coordination, such as Swarm-Based Models and Market-Based Approaches; (ii) comparison of the efficiency of PSO with Genetic Algorithms and Simulated Annealing; (iii) the improvement of the fire simulation model. After the evaluations of these approaches, real robots should be used in the experiments.

ACKNOWLEDGEMENTS

The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology – Critical Embedded Systems – Brazil), processes 573963/2008-8 and 08/57870-9. Also acknowledge Capes for financial support.

REFERENCES

Balch, T. and Arkin, R.C. (1998). Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6), 926–939.

Burchardt, H. and Salomon, R. (2006). Implementation of path planning using ga on mobile robots. In *Proc. of IEEE Congress on Evolut. Computation*, 1831–1836.

Darpa (2007). Darpa grand challenge webpage, www.darpa.mil/grandchallenge.

Demeter (2009). Terrain engine, demeter.sourceforge.net.

Dorigo, M., Trianni, V., Sahin, E., Gro, R., Labella, T., Baldassarre, G., Nolfi, S., Deneubourg, J., Mondada, F., Floreano, D., and Gambardella, L. (2004). Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17, 223–245.

Eberhart, R.C., Kennedy, J., and Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann, San Fransisco, CA.

Eberhart, R.C. and Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. In *Proc. of Int. Conf. on Evolut. Programming*, 611–616.

Eberhart, R.C. and Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In *Proc. of IEEE Congress Evolut. Computation*, 81 – 86.

Engelbrecht, A.P. (2005). *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons.

Go, J., Browning, B., and Veloso, M. (2004). Accurate and flexible simulation for dynamic, vision-centric robots. In *Int. Conf. on Aut. Agents (AAMAS)*, 1388–1389.

Heinen, M., Osório, F., Heinen, F., and Kelber, C. (2006). Seva3d: Using artificial neural networks to autonomous vehicle parking control. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*.

IdMind (2009). Projecto raposa, http://raposa.idmind.pt.

Kennedy, J. (1999). Small worlds e mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proc. of Cong. of Evolut. Computation*, 1931 – 1938.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948. IEEE Press.

Koproski, L.P. (2005). *O Fogo e Seus Efeitos Sobre a Heperto e a Mastofauna Terrestre no Parque Nacional de Ilha Grande*. Master's thesis, UFPR.

Macedo, A.R.M., Macedo, A.R.L., and Duarte, J.B.F. (2007). Robótica aplicada ao combate a incidentes. *Revista TN Petróleo*, (53), 108–113.

Mondada, F., Gambardella, L.M., Floreano, D., and Dorigo, M. (2005). The cooperation of swarm-bots: Physical interactions in collective robotics. *IEEE Robotics and Automation Magazine*, 12, 21–28.

Niehaus, C., Röfer, T., and Laue, T. (2007). Gait-optimization on a humanoid robot using particle swarm optimization. In *Proc. of The Second Workshop on Humanoid Soccer Robots*.

OSG (2009). Open scene graph, osg community website, http://www.openscenegraph.com.

Pessin, G., Osório, F., Hata, A.Y., and Wolf, D.F. (2010). Intelligent control and evolutionary strategies applied to multirobotic systems. In *Proceedings of IEEE-ICIT 2010 International Conference on Industrial Technology*, 1427–1432. IEEE Computer Society.

Pessin, G. (2008). *Evolução de Estratégias e Controle Inteligente em Sistemas Multi-Robóticos Robustos*. Master's thesis, Universidade do Vale do Rio dos Sinos, Available at http://pessin.googlepages.com/disspessin2008.pdf.

Pessin, G., Osório, F.S., Musse, S.R., Nonnenmacher, V., and Ferreira, S.S. (2007). Simulação virtual de agentes autônomos para a identificação e controle de incêndios em reservas naturais. In *Proc. of IX Symposium on Virtual and Augmented Reality (SVR'07)*, 236–245.

Pugh, J. and Martinoli, A. (2006). Multi-robot learning with particle swarm optimization. In *Proceedings of the fifth international joint conference on autonomous agents and multiagent systems (AAMAS'06)*, 441–448. Association for Computing Machinery (ACM).

Rong, C., Wang, Q., Huang, Y., Xie, G., and Wang, L. (2008). Autonomous evolution of high-speed quadruped gaits using pso. In *Robot Soccer World Cup*, 259–270.

SDL (2009). Simple directmedia layer, www.libsdl.org.

Smith, R. (2009). Open dynamics engine, www.ode.org.

Yamaguchi, H. (1997). Adaptive formation control for distributed autonomous mobile robotgroups. In *IEEE Int. Conf. on Robotics and Automation*, 2300–2305.

Zhao, Y. and Collins, E.G. (2005). Robust automatic parallel parking in tight spaces via fuzzy logic. *Robotics and Autonomous Systems*, 51, 111–127.